

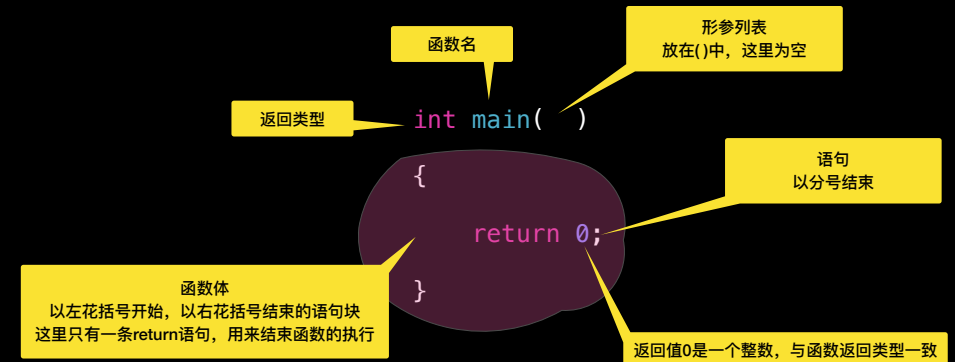
SOEN2070 C++程序设计

01 概述

刘安 anliu@suda.edu.cn 2023-2024-2

1

一个简单的C++程序



2

从命令行编译和运行程序

```
int main( )  
{  
    return 0;  
}
```

使用g++命令来编译链接C++源文件

```
(base) ryan@ryandeMac-Studio 01 % ls  
main_only.cc  
(base) ryan@ryandeMac-Studio 01 % g++ main_only.cc  
(base) ryan@ryandeMac-Studio 01 % ls  
a.out      main_only.cc  
(base) ryan@ryandeMac-Studio 01 % ./a.out  
(base) ryan@ryandeMac-Studio 01 %
```

运行生成的可执行文件

编译链接后生成了一个可执行文件

3

使用iostream标准库提供的输入输出功能

```
#include <iostream> // 通过#include指令使用iostream库  
  
int main() // 表示单行注释  
{  
    // prompt user to enter two numbers  
    std::cout << "Enter two numbers:" << std::endl; // 将信息输出至屏幕  
  
    int v1 = 0, v2 = 0; // 定义两个整型变量保存数据  
    std::cin >> v1 >> v2; // 接受用户输入  
  
    std::cout << "The sum of " << v1 << " and " << v2  
    << " is " << v1 + v2 << std::endl; // 将信息输出至屏幕  
    return 0;  
}
```

4

std::cout和输出运算符<<

- 输出运算符<<
- 二元运算符，左侧必须是ostream对象，右侧是要输出的值
- 将右侧的值写到左侧对象中，计算结果仍然是左侧对象
- std::cout是一个ostream对象
- std::endl称为操纵符，用来结束当前行，并刷新缓冲区

```
std::cout << "Enter two numbers:" << std::endl;  
std::cout << std::endl;
```

5

std::cin和输入运算符>>

- 输入运算符>>
- 二元运算符，左侧必须是istream对象，右侧是一个用来存储值的对象
- 从左侧对象读入数据并存在右侧对象中，计算结果仍然是左侧对象
- std::cin是一个istream对象，其中存放了从键盘输入的数据

```
std::cin >> v1 >> v2;  
std::cin >> v2;
```

6

用while语句表达循环

```
#include <iostream>  
  
int main()  
{  
    int sum = 0, val = 1;  
    while (val <= 10) {  
        sum += val; // assigns sum + val to sum  
        ++val;     // add 1 to val  
    }  
    std::cout << "Sum of 1 to 10 inclusive is "  
              << sum << std::endl;  
  
    return 0;  
}
```

条件放在()内

复合语句/语句块

7

用for语句表达循环

```
int main()  
{  
    int sum = 0;  
  
    // sum values from 1 through 10 inclusive  
    for (int val = 1; val <= 10; ++val)  
        sum += val; // equivalent to sum = sum + val  
  
    std::cout << "Sum of 1 to 10 inclusive is "  
              << sum << std::endl;  
  
    return 0;  
}
```

for (初始化表达式; 条件表达式; 迭代表达式)

1. 首先执行初始化表达式
2. 然后执行条件表达式
 - 2.1. 如果值为真，执行循环体，再执行迭代表达式，然后跳转步骤2
 - 2.2. 否则结束循环

读取数量不定的输入数据

```
int main()
{
    int sum = 0, value = 0;

    // read until end-of-file, calculating
    // a running total of all values read

    while (std::cin >> value)
        sum += value; // equivalent to sum = sum + value

    std::cout << "Sum is: " << sum << std::endl;
    return 0;
}
```

输入运算符返回的还是左侧对象std::cin

std::cin作为条件
true: 成功读取数据, 比如这里是一个整数
false: 没有成功读取数据, 比如遇到文件结束符, 或遇到一个无效输入

演示 mysum.cc

10

统计用户输入的值连续出现的次数

```
while (std::cin >> val) { // read the numbers
    // 要么出现次数+1
    // 要么出现次数=1    关键: 上一个值是否等于val
}
```

在进入循环之前, 上一个值=?

方法: 先读入第一个值, 再进入循环

```
01 -- zsh -- 80x24
(base) ryan@ryandeMac-Studio 01 % ./occurs
42 42 42 42 42 55 55 62 100 100 100
42 occurs 5 times
55 occurs 2 times
62 occurs 1 times
100 occurs 3 times
(base) ryan@ryandeMac-Studio 01 %
```

11

std::cin作为条件

用if语句表达条件执行

```
if (std::cin >> currVal) {
    int cnt = 1; // store the count for the current value we're processing
    while (std::cin >> val) { // read the remaining numbers
        if (val == currVal) // if the values are the same
            ++cnt;          // add 1 to cnt
        else { // otherwise, print the count for the previous value
            std::cout << currVal << " occurs " << cnt << " times" << std::endl;
            currVal = val; // remember the new value
            cnt = 1;       // reset the counter
        }
    } // while loop ends here
} // outermost if statement ends
```

```
1 -- zsh -- 71x24
(base) ryan@RyandeMac-Pro 1 % ./occurs
42 42 42 42 42 55 55 62 100 100 100
42 occurs 5 times
55 occurs 2 times
62 occurs 1 times
(base) ryan@RyandeMac-Pro 1 %
```

100去哪里了?

12

std::cin作为条件

用if语句表达条件执行

```
if (std::cin >> currVal) {
    int cnt = 1; // store the count for the current value we're processing
    while (std::cin >> val) { // read the remaining numbers
        if (val == currVal) // if the values are the same
            ++cnt; // add 1 to cnt
        else { // otherwise, print the count for the previous value
            std::cout << currVal << " occurs " << cnt << " times" << std::endl;
            currVal = val; // remember the new value
            cnt = 1; // reset the counter
        }
    } // while loop ends here
    // remember to print the count for the last value in the file
    std::cout << currVal << " occurs " << cnt << " times" << std::endl;
} // outermost if statement ends here
```

13

使用用户自定义类型

- 使用标准库需要包含相关的头文件，比如#include <iostream>
- 用户也可以设计一个类：定义一个类型以及与其关联的一组操作
- 使用一个类需要了解
 - 类名是什么？
 - 类是在哪里定义的？
 - 它支持什么操作？
- 下面使用类Sales_item
 - 表示一本书的总销售额、售出册数和平均售价

14

使用用户自定义类型Sales_item

```
#include <iostream>
#include "Sales_item.h"
```

来自标准库的头文件，使用<>
来自非标准库的头文件，使用""

```
int main()
{
```

类名

类是在哪里定义的

```
    Sales_item book;
```

```
    // read ISBN, number of copies sold, and sales price
    std::cin >> book;
```

```
    // write ISBN, number of copies sold, total revenue, and average price
    std::cout << book << std::endl;
```

```
    return 0;
```

```
}
```

类支持的操作

类支持的操作

15

确保Sales_item.h与item_io.cc在同一文件夹中

```
(base) ryan@ryandeMac-Studio 01 % ls
Sales_item.h  data  item_io.cc  mysum.cc  whilecount.cc
a.out         forcount.cc  main_only.cc  occurs
add.cc        item_io      mysum        occurs.cc
(base) ryan@ryandeMac-Studio 01 % g++ item_io.cc -std=c++11 -o item_io
(base) ryan@ryandeMac-Studio 01 % ./item_io
0-201-70353-X 4 24.99
0-201-70353-X 4 99.96 24.99
(base) ryan@ryandeMac-Studio 01 %
```

输入：以每本24.99元的价格售出4本书

输出：共售出4本书，总销售额是99.96元，平均销售价格是24.99元

16

使用用户自定义类型Sales_item

```
Sales_item item1, item2;
std::cin >> item1 >> item2;
// first check that item1 and item2 represent the same book
if (item1.isbn() == item2.isbn()) {
    std::cout << item1 + item2 << std::endl;
    return 0; // indicate success
} else {
    std::cerr << "Data must refer to same ISBN"
               << std::endl;
    return -1; // indicate failure
}
```

通过点运算符调用名为isbn的成员函数
左侧是一个类类型的对象，右侧是该类型的一个成员名

通过调用运算符(+)来调用一个函数

将两个Sales_item对象相加
具体含义由类设计者决定
总销售额相加、销售册数相加

17

使用文件重定向简化程序测试

- 测试程序时，反复从键盘输入数据非常乏味
- 使用文件重定向将标准输入和标准输出与文件关联起来

```
(base) ryan@ryandeMac-Studio 01 % ls data
add      add_item      book_sales  mysum      occurs
(base) ryan@ryandeMac-Studio 01 % cat data/add_item
0-201-78345-X 3 20.00
0-201-78345-X 2 25.00
(base) ryan@ryandeMac-Studio 01 % ./add_item < data/add_item
0-201-78345-X 5 110 22
(base) ryan@ryandeMac-Studio 01 % ./add_item < data/add_item > data/add_result
(base) ryan@ryandeMac-Studio 01 % cat data/add_result
0-201-78345-X 5 110 22
(base) ryan@ryandeMac-Studio 01 %
```

显示add_item文件内容

从add_item文件读取数据

将程序结果输出到add_result文件

18